

WAHSP

End-user Manual

Fons Laan
Informatics Institute
University of Amsterdam
Science Park 904
1098 XH Amsterdam

version 0.2.2

8 Jun 2012



Web Application for Historical Sentiment mining in Public media

Contents

| | | |
|----------|---------------------------------|-----------|
| 1 | Introduction | 1 |
| 2 | User interface | 1 |
| 3 | Searching | 3 |
| 3.1 | Query editor | 3 |
| 3.2 | Combining Queries | 6 |
| 4 | Word cloud configuration | 8 |
| 5 | Sentiment highlighting | 10 |
| 6 | Some abbreviations | 12 |
| 7 | Acknowledgments | 12 |

1 Introduction

In this document we will describe how to use the web application of the Clarin WAHSP project. With your browser ¹ you can find the application at <http://dev.wahsp.nl>.

WAHSP is a research tools for historians that uses the newspaper data of the KONINKLIJKE BIBLIOTHEEK as input material. One can search with single query terms or with combinations thereof. Apart from showing the articles that match the query, the results can be visualized by word clouds of single articles together with sentiment words highlighted, or by a word cloud of the whole result set together with newspaper statistics derived from their metadata.

Additional information about the project can be obtained from the BILAND CMS site <http://biland.nl>, which is the successor of WAHSP.

2 User interface

In this section we will give an overview of the components of the user interface. After accessing the WAHSP URL you will see the login window, see fig. 1. Just clicking the **Login** button makes you a guest user, but WAHSP collaborators will use their own credentials. Notice that there is only one guest account, so other guests can delete the queries that you —as a guest— saved.

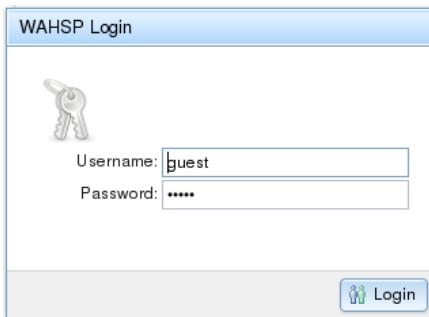


Figure 1: Login window.

The WAHSP opening window is shown in fig. 2.

¹Internet Explorer may not work with WAHSP. Please use Google Chrome, a recent Firefox, Opera, ...

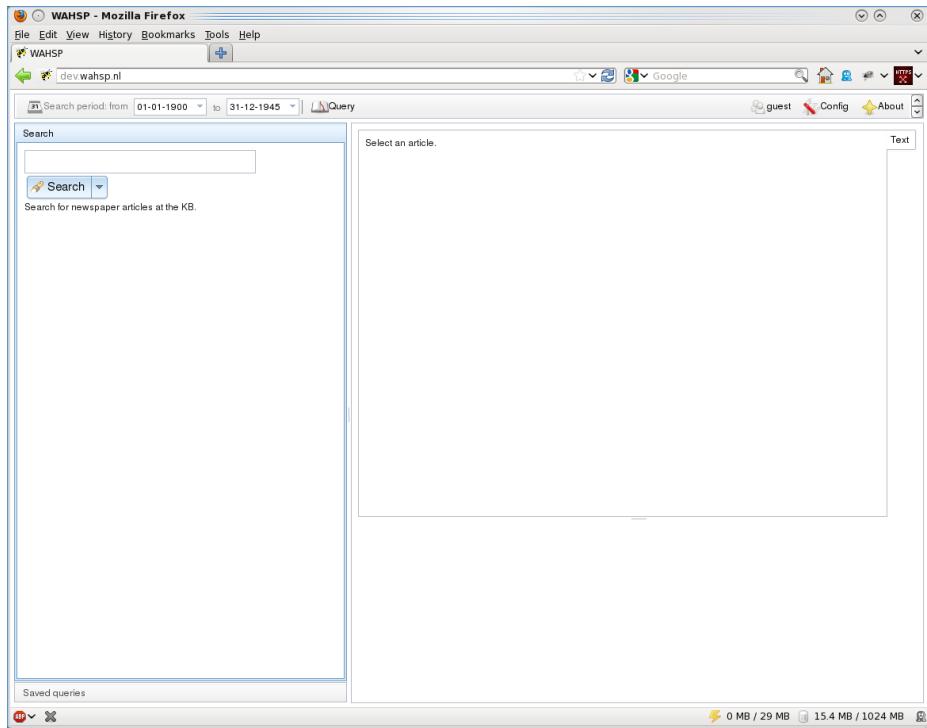


Figure 2: WAHSP opening screen.

The window consists of the following screen regions:

- The toolbar at the top
- An accordion widget at the left
- The article tab widget at the top-right
- A region for the word cloud at the bottom-right

The *toolbar* consists of the following widgets:

- Two date widgets to limit the search period. The full date range of the KB newspapers is 1900–1945.
- A query widget, used to combine saved queries into a new query.
- A logout widget.
- A configuration widget, mostly for word cloud options.
- An about widget, showing the collaborators of the project, and a link to this document.

The query *accordion* on the left has the two divisions:

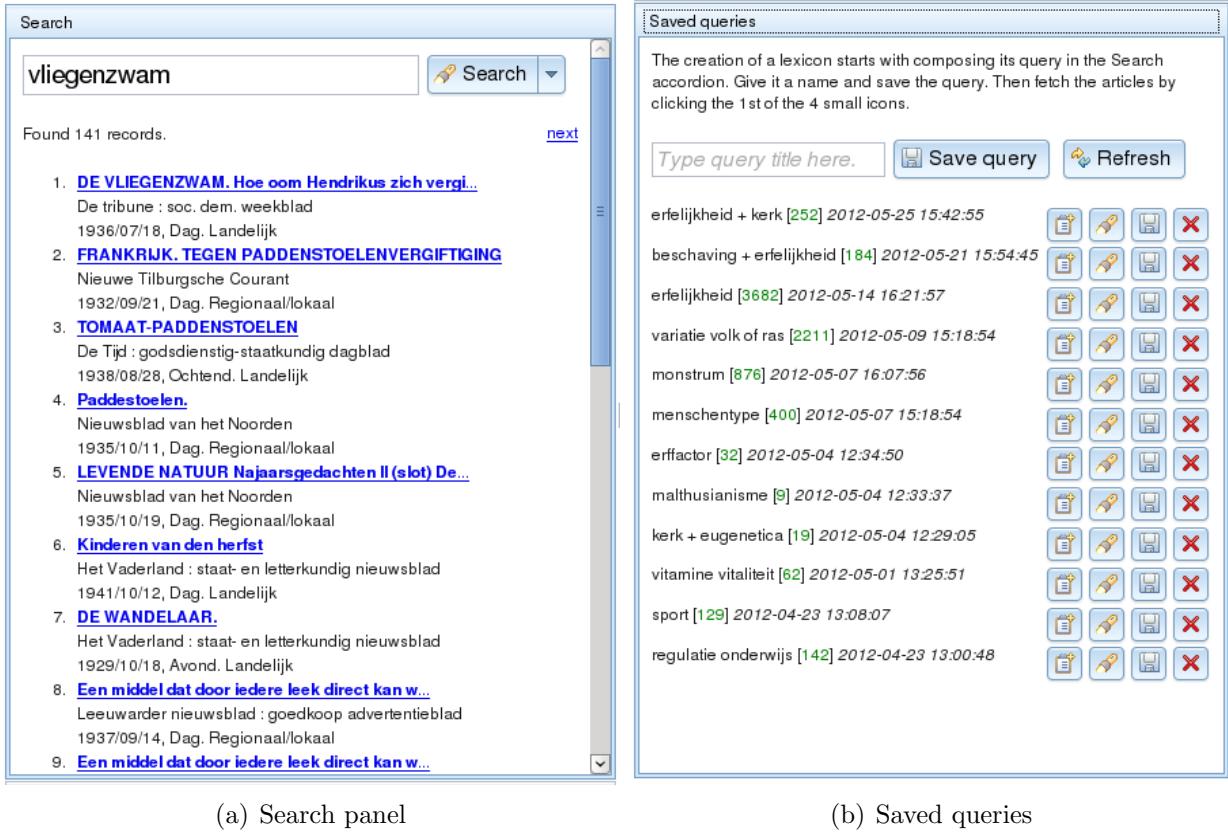
- **Search**
Here one creates new queries, to be sent to the KB search engine.
- **Saved queries**

This shows the list of your saved queries, which are used to retrieve the OCR data of the articles, create word clouds, and display newspaper statistics.

The screen area to the right of the accordion is for displaying the OCR, statistics and clouds, and will be discussed together with searching.

3 Searching

A trivial way to search is by using a single query word. Say, we type **vliegenzwam** in the textline area in the accordion, and then click the **Search** button; see fig. 3a. It shows that 141 articles are found. The first chunk is displayed with their titles in blue and underlined. Underneath the title is some additional information: the newspaper title, article date, and newspaper ‘type’ (country-wide, or regional). Clicking next gives the next chunk of articles.



(a) Search panel

(b) Saved queries

Figure 3: Search and Saved queries in the accordion.

When you click one of the article titles, its OCR text is shown in the **Text** tab, see fig. 4. Clicking the **Original** tab shows the scan image of the newspaper article (fig. 5).

The third tab **View at KB** opens the KB search engine page in a new browser window (or tab).

The corresponding word cloud of the article is shown in fig. 6. The used font size of the words is the graphical equivalent of their frequency in the document. Words of too low frequency may not be shown, and in general ‘noise’ is also suppressed.

Inspecting the words in the cloud may lead one to make adaptations to the original query.

3.1 Query editor

Creating queries that consist of more than a single word is done with the built-in query editor. The editor is easiest to explain by creating an example query. Let say that we create a new query that we will later save with the name ‘luminal’. Proceed with the following steps:

- In the **Search** panel of the accordion, type **luminal** as search term.
- Click on the tiny arrow on the right half of the **Search** button.
- Click on the button **Start search** that appeared underneath the **Search** button.

LEVENDE NATUUR Najaarsgedachten II (slot) De „Sporenfoto”.

Op mijn tafel staat weer, zoals iederen herfst, een schaaltje opgemaakt met paddenstoelen, mos en herfstvruchten en daarvan heb ik dagenlang genoegen. Voorzichtig ingepakt tusschen wat gras in onze botaniseertrommels hebben we ze meegenomen en als morgen studie- of andere bezigheden ons thuis houden, hebben we toch een klein, bescheiden stukje herfstbosch in onze kamer, waaraan we ons bovenmatig kunnen verwikkelen en dat zoete herinneringen wekt. De hoed van de uitgegraven vliegenzwam hebben we zonder stiel, plat op een zwart carton gelegd op een windstille plek — in een tast — en daar vormt zich nu in den loop van den nacht een prachtige z.g. „sporenfoto”. De sporen namelijk, welke bij miljoenen uit de plaatjes aan den onderkant van den hoed vallen, rangschikken zich op het carton in 'n figuur welke met de groepeerind der plaatjes overeenkomt. De sporen der vliegenzwam zijn wit. De zwavelkopjes hebben purperbruine sporen, zoodat we die op wit papier hebben gelegd. De gekleurde sporenfiguren fixeeren we, door er van onderen met een penseeltje fixatief tegen te strijken. Deze dringt dan door het papier heen en fixeert de sporen 'aan de bovenzijde. Voor witte sporen die door deze bewerking onzichtbaar worden, is ons geen goede methode bekend. Alvorens van de paddenstoelen af te stappen, wil ik nog even vermelden, dat mevrouw Kraus ons in de Vosbergen een prachtige sponszwam — *Sparassis crispa* — heeft gewezen, een vrij zeldzame soort, die we er tot nu toe nooit hebben aangetroffen.

Figure 4: OCR text of a KB article in the **Text** tab.

FRANKRIJK.
TEGEN PADDENSTOELENVERGIFTINGEN

De corr. van de N. R. C. te Paris meidt:
Vóór enige weken was er al sprake van een door een Fransch geneesheer gevonden middel ter bestrijding van vergiftiging door het eten van paddenstoelen. Dr. Limousin heeft er tegen iemand van de Matin over gesproken. Tot nu toe was er geen enkel rationeel middel om het vergif van de *vliegenzwam* te neutraliseren. De wetenschap moet er zich meer vergenoegen elk verschijnsel afzonderlijk aan te pakken en het organisme in den ongelijken strijd te stellen.

Men wist al lang, dat het konijn ongestraft van die zwam kan eten, maar er spoedig aan sterft, als men het vocht onder de huid inspuit. Men zag daarom slechts een verschijnsel, te vergelijken met slangenbit. Dr. Limousin bediende zich voor zijn proef van een tweede dier, de kat, die zich een schaaltje gekookte *vliegenzwam* uitstekend liet smaken. Een konijn, dat van nature re geimmeerde is, verlikt het daarentegen ook maar iets van de portie, gekookt of rauw, te verorberen. Dr. Limousin gaf toen de kat, tegelijk met de *vliegenzwam* iets van de maag van een konijn, waarna ze aan *Il* soort progresieve verlamming bezweek. Hieruit concludeerde daaruit, dat er twee vergiftigen in het spel zijn, het een geneutraliseerde door de konijnenmaag, dat gewoonlijk doodt en een ander, dat langzaam werkt en voor namelijk het zenuwstelsel aantast. Hier vloeide toen bij de *vliegenzwam* behalve maag- ook hersenstof van een konijn.

De kater die daarvan al heeft de proef schitterend doorstaan. En zwarte vergiftiging bij zes personen, die ze anders met de dood hadden moeten bekoopen, zijn, aldus het verslag in de Matin, door die geneesmiddelen gered.

Figure 5: Scan image of a KB article in the Original tab. The word **vliegenzwam** is blue because that was the query.



Figure 6: Word cloud of a single KB article.

- Below the text widget (that now contains (cql.serverChoice exact "luminal")) there is a new button with text **luminal**. Click on its arrow at the right side.
- You will see a new frame with several buttons and other widgets. Click on the button **Make word list**.
- Next to **Word list: luminal** there is a tiny icon of the inline editbox, click on it.
- Type **chloral** in the text region (see fig. 7) en then press Enter. Next to **luminal** we now also see **chloral** in the word list.
- Once more press the icon.
- Type **wekaminen** and press Enter.
- Click the **Search** button, which shows the found records.
- Then go to **Saved queries** in the accordion and at **Type query title here**. type **luminal**, and click the **Save query** button. Then **luminal** is displayed as the new saved query (unless that name is already taken).
- Click its first icon (with hover text **Create basis lexicon: luminal**). This loads the OCR data of all the luminal articles from the KB.
- When the loading is done, click the second icon **Apply query: luminal**. That creates the cloud of the luminal articles, plus some statistics.

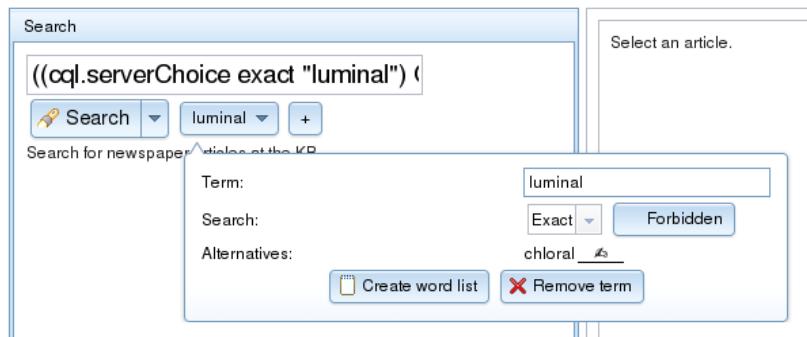


Figure 7: Query editor.

The **Saved queries** panel shows the query titles, their article count, and the creation date of the queries that you have saved. If the article count is zero, either you have not loaded the KB data, or there just is no loadable data, because your query did not yield a single hit. To the right of each query are four small icons. When you move your mouse over them, you will see their hover text:

- **Create basis lexicon**
- **Apply query**
- **Modify**
- **Delete**

After you saved a new query, it is important to realize that you cannot show the word cloud of those articles together yet, because the OCR text of all articles has to be fetched from the KB, and be pre-processed by our xTAS (Text Analysis Service), see <http://xtas.net>. That will be accomplished by clicking the first of the four icons. When the query yielded many articles it is time for coffee.

After a while the loading is done (fig. 8), and the number of articles is shown. Please remember this number for moment. What is actually done, is that WAHSP finished delegating all the hard work to a bunch of helper processes. And they may need a bit more time.

The new lexicon now appears in the accordion. The number after the lexicon name in brackets shows the number of articles available. If it is a **single number** identical to the number mentioned before, then the loading is done. But it may easily happen that you see **two numbers** which are the separate counts of the article metadata and OCR. It likely means that the WAHSP helpers are still busy. You may click the **Refresh** button to see if progress is being made. When the metadata and OCR counts are non-zero the second tiny icon will have been enabled, and you can proceed to look at preliminary word clouds and graphs of statistics.

There are two other reasons that may lead to article counts changing over time.

- The KB digitization of the historical newspapers is still an ongoing process. Once in a while new data is made available. WAHSP does not check this, but when you manually reload the data you may see an increase of the number of articles.
- Another issue is that over time (days, weeks) the metadata and OCR count may become different. This is an unresolved bug (like WAHSP itself?). Reloading the data will fix this, at least for some time.

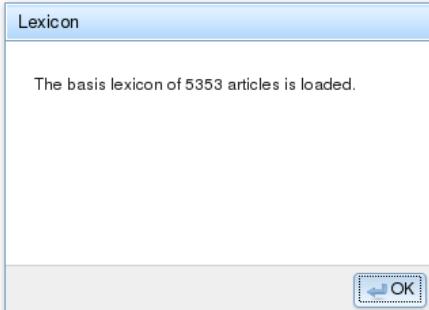


Figure 8: Loading of the KB data seems to be done.

So the article set corresponding to the query must first be loaded in order to view the cloud of all words together. For a single article you can view the word cloud immediately. The cause of this difference is that with a single article the fetching of the data is done automatically.

When fetching and pre-processing the articles is done, you can click the second icon, which now produces the word cloud of all articles together (after a while, accumulating all the word frequencies), and some basic statistics of the lot in the text panel, see fig. 9.

3.2 Combining Queries

With the query widget (see fig. 10, reachable from the toolbar) one can combine two existing (i.e. saved) queries into a new query. First select the desired boolean combination operator (**AND**, **OR** or **NOT**), and then select the first and second query from the available list. The widget will suggest a name for the combined query, but you can change that before clicking **OK**.

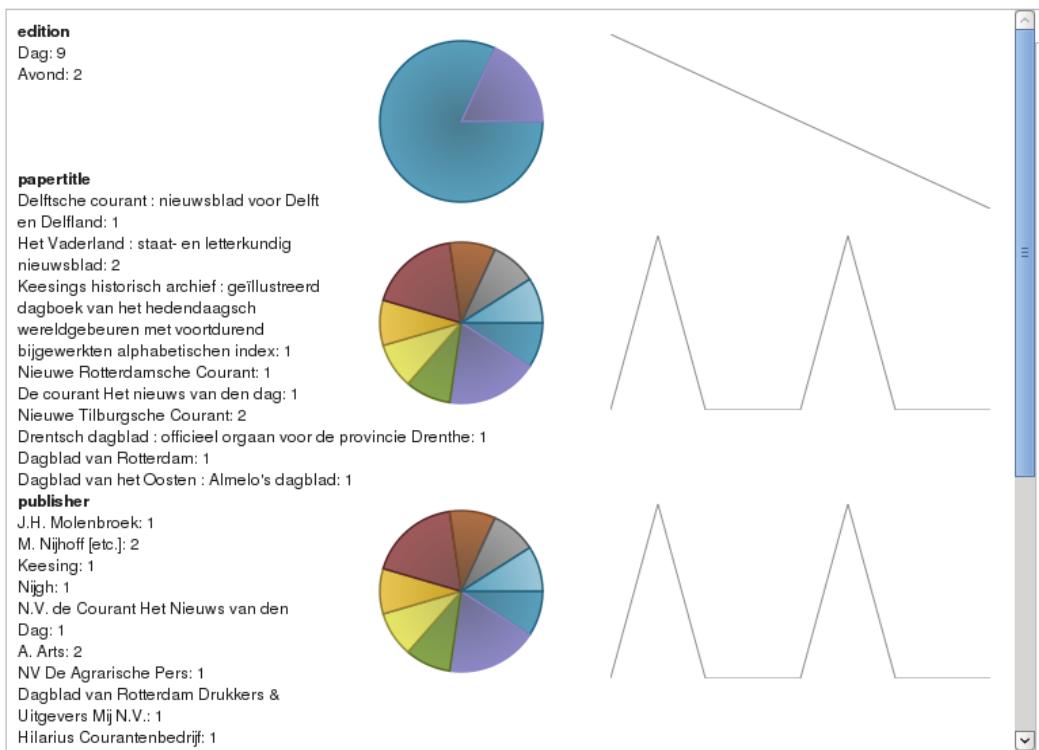


Figure 9: Some newspaper statistics corresponding to the query. Hovering the pie pieces displays the corresponding text.

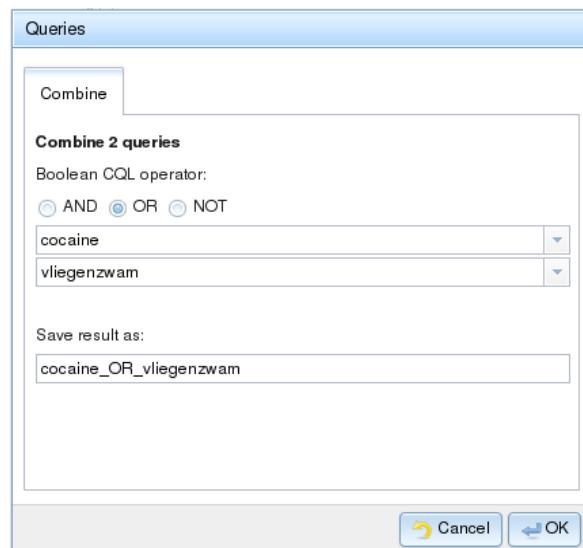


Figure 10: Combining existing queries.

4 Word cloud configuration

The word cloud in fig. 6 was made with default cloud parameters, but there are several options to tune the result according to your wishes. Fig. 11 shows the word cloud options. This configuration widget can be opened from the toolbar.

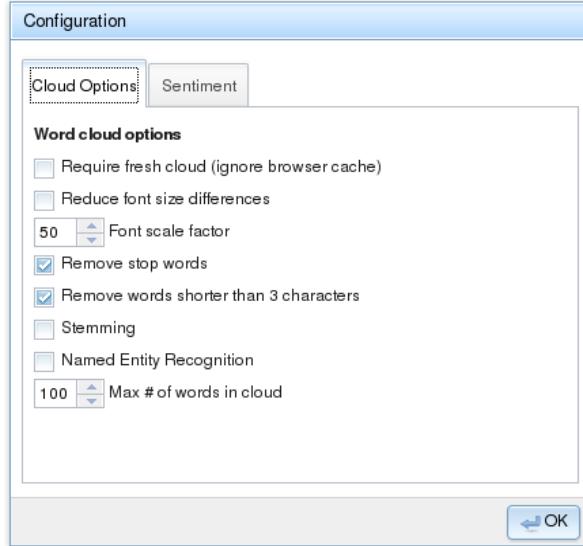


Figure 11: Word cloud configuration.

The word cloud options have the following effect:

- **Require fresh cloud.** This adds a dummy variable with random value to the cloud request. This should convince your browser not to return a cached result.
- **Reduce font size differences.** When the word sizes decline too fast at the cloud edge, this option should improve the result.
- **Font scale factor.** This scale factor determines the maximum font size.
- **Remove stop words.** This removes short words, as specified by a pre-defined list.
- **Remove words shorter than 3 characters.** When the stop word list does not block enough noise, this will filter more.
- **Stemming.** This applies stemming to the words before computing the cloud.
- **Named-Entity Recognition.** This applies NER, currently a bit slow.
- **Max. # of words in cloud.** The number of words returned by the server can be very big. Truncating the list before generating the cloud speeds it up.

Fig. 12 shows the word cloud of the query `wekaminen`, which yields (only) 11 articles. Often, as in this case, the cloud does not properly occupy the available space. One can increase the maximum number of words displayed to remedy this, assuming more words are indeed available. But when the words at the border of the cloud are already small, that does not help much, because words that are too small become invisible anyway. Then it is better to reduce the font size differences, see fig. 13 for the result.

Finally, fig. 14 shows the same word cloud with Named-Entity Recognition. Used colors: `locations`, `persons`, `organizations`, and `miscellaneous`. The latter means that the NER algorithm ‘thinks’ these are entities, but cannot be more specific about it. The NER we used is Stanford, trained for Dutch. It is not perfect, but it is better than several alternatives. Notice that the figure only shows the recognized entities, the remaining words are left out.



Figure 12: Word cloud of the query `wekaminen`.



Figure 13: Word cloud of the query `wekaminen` with reduced font size differences.



Figure 14: Word cloud of the query *wekaminen* with *Stanford* Named-Entity Recognition. Used colors: **locations**, **persons**, **organizations** and **miscellaneous**.

5 Sentiment highlighting

In fig. 4 we showed the plain OCR text of an article. After turning on the sentiment option in the configuration widget (see fig. 15), the article OCR looks as depicted in fig. 16, with **positive** and **negative** sentiment words highlighted.

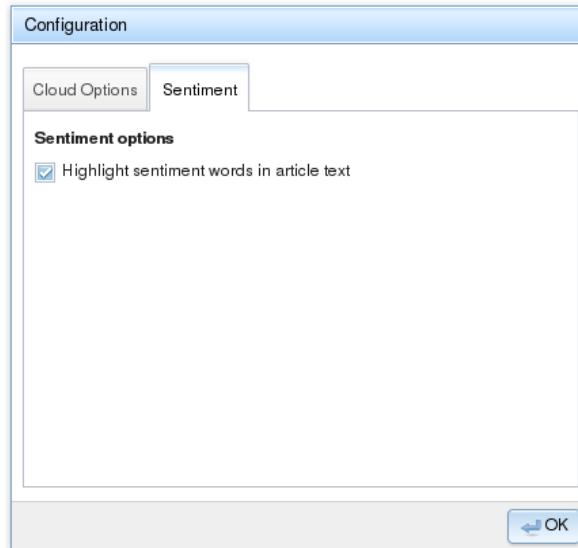


Figure 15: Sentiment option in configuration widget.

This is an article from the query **monstrum**. (Now that we speak of sentiment, should we add **monstrum** to the red list?) The figure also shows that what is highlighted are not whole words, but substrings, which may lead to curious mistakes. And the OCR will never be perfect, which clearly affects the results².

²Apart from OCR mistakes, there is a second shortcoming in the data. The semi-automatic segmentation of the newspaper scans into individual articles is not perfect either, leading to numerous ‘oversegmentation’: ‘articles’ consisting of just their title, their body text having been delegated to the next article. The current settings of the KB search engine imply that short articles come first in the result list.

The corresponding article cloud and scan are shown in figs. 17 and 18.

Een kat in kapok!

G. R. Weer gewaagt de pers van een erger» lijke dierenmishandeling, erger dan die van den boer, die zijn paard met een hooivork ten **dood** mishandelde. Een n.b. 18-jarige **lummel**, die **voor** verzending kapok moest pakken, deed daarin een levende kat, met 't gevolg dat de ontvanger van 't pak, na maanden het openend, daarin de **dood** kat vond die, blijkens de omgeving, hevig te keer moest zijn gegaan **voordat** de verlossende **dood** aan het **lijden** een einde maakte....

Ziet eens hier: Zoon individu, dat in koelen bloede, **voor** de **grap**, dat deed, kan nu mogelijk 6 maanden krijgen, maar wat is dat in vergelijking met het **lijden** van dat dier?

Men kan **tegen** lijfstraffen zijn, maar bij zoo leis gaat men onwillekeurig. In de eerste opwelling, spontaan toch leis voelen **voor** een jus tallonis. Men krijgt 't gevoel om zoon ondier te lijf te gaan en te **pijnigen**. Maar bij nadere overweging is, praktisch bekeken, zoon monstrum «en gevaarlijke sadist. «en...psvchopaath» al, men **wil**, die niet gestraft moet worden (want dat geeft toch niets), maar preventief „ler beschikking“ gesteld, **levenslang** als gedegenererde geëlimineerd moet worden, eclipseren. Het is een monstrum dat, **voor** een lolletje brandsticht, kinderen mutilert en verkracht, en dergelijke meer. Verbetering is uitgesloten: er is een morele missing link, die niets is bij te brengen. übllmeeren (figuurlijke vergassing) is 't enige' blijvend! — Tabula rasa! Rouville.

Text
Original
View at KB

Figure 16: OCR text of the KB article *Een kat in kapok!* with positive and negative sentiment highlighting.



Figure 17: The word cloud of the newspaper article *Een kat in kapok!*.

Een kat in kapok!

G. Wehr gewaagt de pers van een ergelijke dierenmisshandeling, erger dan die van den boer, die zijn paard met een hooivork doodde mishandeld. Een n.b. 18-jarige lummel, die voor verzending kapok moest pakken, deed daarin een levende kat, met 't gevolg dat de ontvanger van 't pak, na maanden het openend, daarin de doodle kat vond die, blijkens de omgeving, hevig te keer moest zijn gegaan voordat de verlossende dood aan het lijden een einde maakte...

Ziel eens hier: Zoo'n individu, dat in koelen bloede, voor de grap, dat deed, kan nu mogelijk 6 maanden krijgen, maar wat is dat in vergelijking met het lijden van dat dier?

Men kan tegen lijfstraffen zijn, maar bij zoo iels gaat men onwilligeurig, in de eerste opwelling spontaan toch iets voelen voor een jus talonis. Men krijgt 't gevoel om zoo'n ondier te lilt te gaan en te pijnigen. Maar bij nadere overweging is, praktisch bekijken, zoo'n monstrum een gevvaarlijke sadist, een „psychopath" als men wil, die niet gestraft moet worden (want dat geelt toch niets), maar preventief „ter beschikking" gesteld, levenslang als gedegenererde geëlimineerd moet worden, eclipsaeren.

Het is een monstrum dat, voor een lolletje brandsticht, kinderen mutilert en verkracht, en dergelijke meer.

Verbetering is pilgelozen: er is een morele missing link, die niets is bij te brengen.

Sublimeeren (figuurlijke vergassing) is 't enige; blijvend! — *Tabula rasa!*

Rouville.

Figure 18: The scan of the newspaper article *Een kat in kapok!*. The word **monstrum** is blue, because that was the query.

6 Some abbreviations

| Abbr. | Meaning |
|-------|---|
| CQL | Contextual Query Language |
| GUI | Graphical User Interface |
| KB | Koninklijke Bibliotheek |
| NER | Named-Entity Recognition |
| OCR | Optical Character Recognition |
| SRU | Search/Retrieval via URL |
| XML | eXtensible Markup Language |
| xTAS | Text Analysis Service |
| URL | Uniform Resource Locator |
| WAHSP | Web Application for Historical Sentiment mining in Public media |

Table 1: Abbreviations.

7 Acknowledgments

Apart from having received comments from my WAHSP colleagues (DAAN ODIJK, STEPHEN SNELDERS & TOINE PIETERS), I also got contributions from JOSÉ DE KRUIJF and JAAP VERHEUL of Utrecht University, and my new Biland colleague PIM HUIJNEN.